Massive Networks – Visualizing Very Large-Scale Graphs in Immersive Environments

Daniel Filonik
UNSW EPICentre, CSIRO Data61
Sydney, NSW, Australia
d.filonik@unsw.edu.au

Dominic Branchaud
UNSW EPICentre
Sydney, NSW, Australia
d.branchaud@unsw.edu.au

Robert Lawther
UNSW EPICentre
Sydney, NSW, Australia
r.lawther@unsw.edu.au

Piotr Szul
CSIRO Data61
Sydney, NSW, Australia
piotr.szul@data61.csiro.au

Alex Collins
CSIRO Data61
Sydney, NSW, Australia
alex.collins@data61.csiro.au

Tomasz Bednarz
UNSW EPICentre, CSIRO Data61
Sydney, NSW, Australia
tomasz.bednarz@siggraph.org

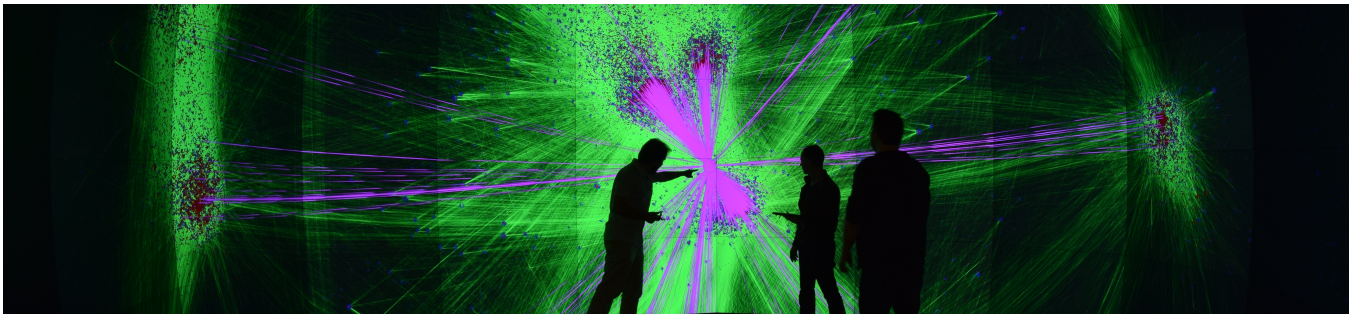

Figure 1: Exploring a large-scale graph in an immersive environment.

ABSTRACT

This work presents our strategy and pipeline architecture for visualizing very large-scale graphs in an immersive environment, using high-performance graphics approach. Innovation lies in utilizing GPUs for real-time cluster-based interactive rendering, but also intermediate graph representation that utilizes Khronos Group's GLTF file format, and interaction design.

CCS CONCEPTS

•**Human-centered computing** → **Graph drawings;** *User interface programming;*

KEYWORDS

immersive visualization, graph visualization, interaction design

ACM Reference format:
Daniel Filonik, Dominic Branchaud, Robert Lawther, Piotr Szul, Alex Collins, and Tomasz Bednarz. 2018. Massive Networks – Visualizing Very Large-Scale Graphs in Immersive Environments. In *Proceedings of High-Performance Graphics, Vancouver, Canada, August 2018 (HPG'18),* 2 pages.
DOI: 10.475/123_4

HPG'18, Vancouver, Canada

DOI: 10.475/123_4

1 INTRODUCTION

A considerable body of work exists in the field of graph visualisation – and a number of mature tools have been developed in various domains, such as *Gephi* [Bastian et al. 2009] or *Cytoscape* [Smoot et al. 2010]. However, most of these tools are designed to handle small to medium sized graphs, with thousands of nodes and edges. In contrast to this, our research aims to develop methods for visualising graphs ranging from one million up to one billion nodes and edges. This roughly corresponds to the maximum amount of memory available on the high-end consumer graphics cards[1], representing an upper bound for fully disaggregated graphs that can still be interactively processed on current hardware.

With the dawn of big data science, such large-scale graph data sets are increasingly common. The potential applications of these interfaces span across a variety of research areas, such as: Biology ('omics), Healthcare, Finance, or Cybersecurity. However, it is also well-documented that large graphs present some serious visualisation challenges, such as the tendency to produce unreadable hairballs and suffer from overplotting. Although they are difficult to resolve in general, some of these issues can be mitigated using modern graphics capabilities. While viewing the whole graph in a top-down approach can be informative, it often makes sense to decimate the graph before proceeding with analysis.

As stated by [Von Landesberger et al. 2011], solving complex analytical tasks concerning large data sets often requires a collaboration of multiple domain experts. However, the authors contend that such interfaces are still under-represented in graph literature.

[1] NVidia GeForce TITAN X/XP/V (12GB)

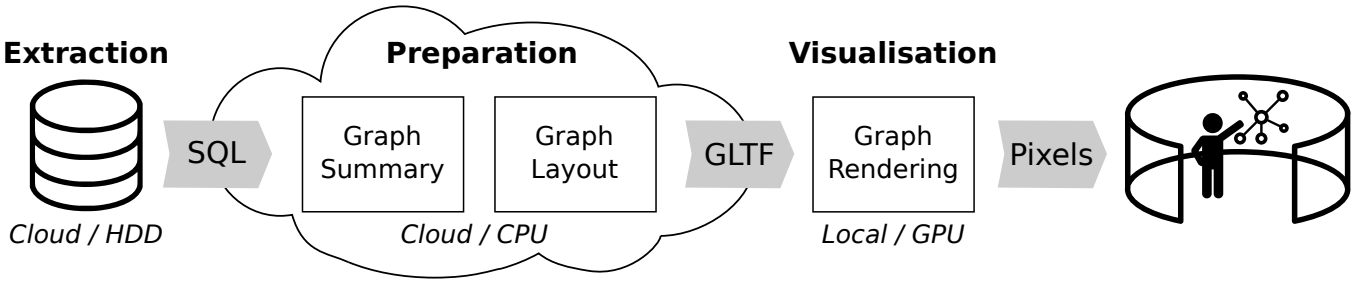

Figure 2: Schematic representation of graph processing pipeline.

The immersive environment used in this project naturally lends itself to collaborative graph exploration. The *EPICylinder* – a very high-resolution cylindrical cave comprised of 56×60" screens with a combined resolution of 26,880 × 4,320 pixels – comfortably accommodates co-located groups of 5-10 analysts (see figure 1).

2 LARGE GRAPH PROCESSING PIPELINE

The following outlines the processing pipeline (see figure 2), highlighting several novel contributions, such as extracting graphs from relational databases, processing them offline or in the cloud, and finally representing them in GLTF format for quick loading and transfer to GPU memory.

Graph Extraction. Any non-trivial database instance of a given schema will generally contain multiple graphs. In particular, given an entity relationship model, the available graphs are identified as follows: Every entity corresponds to a set of nodes, and every relationship between two entities corresponds to a set of edges. Consequently, we treat every database instance as a collection of heterogeneous graphs. The graph extraction stage concludes with the original data converted into an intermediate graph representation suitable for subsequent algorithmic processing.

Graph Preparation. To detect the community structure of the extracted graphs, we adopt the state-of-the-art *Louvain* method [Blondel et al. 2008], which has proven computationally suitable even when large-scale graphs are involved. Subsequently, the following stage is concerned with spatially laying out the graph. The previous algorithms work purely on the abstract graph structure, without assigning concrete positions to nodes or edges. For the purpose of generating layouts, we utilise the *ForceAtlas2* and *OpenOrd* algorithms [Jacomy et al. 2014; Martin et al. 2011].

The preparation stage concludes by converting the graph into the GLTF format [Khronos Group 2012]. The nodes are represented as GL_POINTS, and the edges are represented as GL_LINES. The binary representation is much more compact than textual graph formats commonly in use today, and it maps directly to the representation on the graphics card, resulting in fast loading times even with very large graphs.

Graph Visualisation. The final stage is concerned with the graphical representation and interactive manipulation of the graph. At this stage, the entire graph is loaded into GPU memory. To display the massive networks, we adopt GPU-based point-cloud rendering techniques, producing large numbers of node and edge geometries on-the-fly with the aid of geometry shaders. The use of shaders also allows users to interactively adjust the visual appearance of the graph to aid the analysis (animated transitions or view distortions in real-time).

While hovering the cursor over the display, a short information is displayed in the form of a label. Furthermore, it is possible to retrieve detailed information including all data attributes with an additional button press. A quad tree is used to efficiently detect the closest node to the cursor. To address the problem of information overload and overplotting, it is critical to provide effective mechanisms for reducing the complexity of the graph. In this regard, our system supports three complementary ways of extracting sub-graphs: a) selection based on *graph layout* (geometry), b) selection based on *graph summarization* (hierarchy), and c) selection based on *data attributes* (filtering). Once a sub-graph is chosen, the user can obtain aggregated statistics about the nodes and edges, or continue exploration by further refining the sub-graph.

3 FINAL REMARKS

As described, our solution allows users to visualize very large-scale graphs in an immersive environment. It supports fully interactive exploration and manipulation – and it can co-locate 5-10 analysts in the same space to discuss the outcomes of the visual analysis. In future work, we will apply this system in the area of Metabolomics to analyze complex networks of chemical interactions.

REFERENCES

Mathieu Bastian, Sebastien Heymann, Mathieu Jacomy, and others. 2009. Gephi: an open source software for exploring and manipulating networks. *Icwsm* 8 (2009), 361–362.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. 2008. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* 2008, 10 (2008), P10008.

Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. 2014. ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. *PloS one* 9, 6 (2014), e98679.

Khronos Group. 2012. glTF – The runtime asset format for WebGL, OpenGL ES, and OpenGL. (2012). https://github.com/KhronosGroup/glTF

Shawn Martin, W Michael Brown, Richard Klavans, and Kevin W Boyack. 2011. OpenOrd: an open-source toolbox for large graph layout. In *Visualization and Data Analysis 2011*, Vol. 7868. International Society for Optics and Photonics, 786806.

Michael E Smoot, Keiichiro Ono, Johannes Ruscheinski, Peng-Liang Wang, and Trey Ideker. 2010. Cytoscape 2.8: new features for data integration and network visualization. *Bioinformatics* 27, 3 (2010), 431–432.

Tatiana Von Landesberger, Arjan Kuijper, Tobias Schreck, Jörn Kohlhammer, Jarke J van Wijk, J-D Fekete, and Dieter W Fellner. 2011. Visual analysis of large graphs: state-of-the-art and future research challenges. In *Computer graphics forum*, Vol. 30. Wiley Online Library, 1719–1749.